Complexiteit

Uitwerking Opgave 20

a. $\#(1) = 1$, $\#(2) = \#(3) + 1$, $\#(3) \geq 1$ omdat $n > 1$ (dus de loop wordt minstens één keer uitgevoerd), $\#(4) \leq \#(3)$, $\#(6) \leq \#((3)$, $\#(7) = \#(8) \leq \#(6)$, $\#(10) = \#(6)$.
Dus (I) en (II) zijn allebei maatgevend voor de complexiteit. Operatie (II) is echter de beste basisoperatie, want de werking van het algoritme is gebaseerd op de uitslag van de tests of $A[i]$ resp. $A[j]$ even is, dus (II) is fundamenteel voor de werking. We sorteren immers op even/onevenheid.
Operatie (I) is een vergelijking tussen indices en doet niets met het array; het is meer een boekhoudoperatie dan fundamenteel voor de werking van het algoritme.
Operatie (III) is niet maatgevend. Er zijn immers invoerrijtjes waarvoor 0 verwisselingen plaatsvinden, maar $n - 1$ tests uit regel (3). Bijvoorbeeld rijtjes met alle $A[i]$ even. Geen goede maat dus.

b. In elke ronde wordt ofwel i met 1 opgehoogd, ofwel j met 1 afgelaagd of beide. Dat laatste levert het best case aantal tests (uit regel (2)), want dan wordt het bekeken interval steeds met 2 kleiner. Best case vindt dan ook plaats als in *elke* stap de test in regel (3) False oplevert ($A[i]$ is oneven) en de test in regel (6) True ($A[j]$ is even). In dat geval worden $A[i]$ en $A[j]$ verwisseld en zowel i opgehoogd als j afgelaagd. Elke keer moet dus $A[i]$ oneven en $A[j]$ even zijn, te beginnen bij $i = 1$ en $j = n$, dan $i = 2$ en $j = n - 1$, etcetera. Ergo, de eerste helft moet uit oneven getallen bestaan, de tweede helft uit even getallen.
Iets preciezer: (i) voor n even is in de laatste stap (middelste twee elementen) $i = \frac{n}{2}$ en $j = \frac{n}{2} + 1$. Het maakt dan niet uit of de twee middelste elementen even of oneven zijn; er wordt daarna altijd gestopt omdat $i \geq j$ is geworden. We hebben dan $\frac{n}{2}$ iteraties, dus $\frac{n}{2} + 1$ keer regel (2). (ii) Voor n oneven moeten de eerste $\frac{n-1}{2} = \lfloor \frac{n}{2} \rfloor$ waarden oneven zijn en de laatste $\frac{n-1}{2}$ even, maar de middelste doet er niet toe; zodra $i = j$ stopt het algoritme. In dit geval: $\frac{n-1}{2}$ iteraties, dus $\frac{n-1}{2} + 1 = \frac{n+1}{2} = \lceil \frac{n}{2} \rceil$ keer regel (2).
Samengevat is dit voor algemene n gelijk aan $\lceil \frac{n+1}{2} \rceil$.

In de worst case wordt altijd ofwel i met 1 opgehoogd, ofwel j met 1 afgelaagd, maar nooit beide. Dit betekent dat in elke ronde ofwel $A[i]$ even is, ofwel $A[i]$ oneven *en* $A[j]$ oneven. Er wordt in de worst case dus nooit gewisseld, en het komt nooit voor dat je links (i) een oneven getal tegenkomt en rechts (j) een even getal (behalve eventueel in de laatste ronde, als $j = i + 1$). Dus zodra je met i het eerste oneven getal tegenkomt (en $i < j$) moet er rechts (j) ook een oneven getal staan, en vervolgens mag je -voor de worst case- alleen nog met j naar links lopen, waarbij je alleen nog maar oneven getallen mag tegenkomen[1]. Immers, $A[i]$ was oneven en i is niet veranderd. Dit kan alleen optreden als A er als volgt uitziet: een aantal —eventueel 0— even getallen, gevolgd door een aantal —eventueel 0— oneven getallen. Alle elementen staan dus al op de juiste plaats. Het aantal stappen (tests regel (2)) dat je zo doet is dan uiteraard n, omdat je bekeken interval telkens maar 1 stap kleiner wordt en je stopt zodra je merkt dat i gelijk is aan

[1] behalve eventueel op positie $i + 1$, zie verderop

j (of $\geq j$ in het uitzonderingsgeval). Die uitzondering komt voor als je met i het eerste oneven getal tegenkomt terwijl $j = i + 1$. Het kan verder alleen nog voorkomen als i op het eerste oneven getal staat en j langs oneven getallen teruggelopen is tot $j = i+1$. Dan mag $A[j]$ even zijn. In die twee gevallen mag je wel verwisselen en zowel i als j ophogen, aangezien je sowieso vervolgens stopt. Na verwisselen wordt nog één keer de test in regel (2) gedaan, in totaal dus ook n keer.
Deze aparte gevallen kunnen alleen voorkomen als (i) $A[1]$ t/m $A[n-2]$ even zijn en dan $A[n-1]$ oneven en $A[n]$ even of (ii) direct na het eerste oneven getal vanaf links een even getal staat, en daarachter alleen oneven getallen. Overigens is (i) een speciaal geval van (ii). Voorbeeldrijtje: $2, 4, 8, 3, 2, 7, 5, 9$.
c. Er wordt alleen verwisseld als geldt dat $A[i]$ oneven is *en* $A[j]$ even (*). In dat geval wordt i met 1 opgehoogd, en j met 1 afgelaagd. Het beste geval treedt op als (*) nooit voorkomt, d.w.z. als in elke ronde (doorgang door de while) ofwel $A[i]$ even is, ofwel $A[i]$ en $A[j]$ beide oneven. Zodra we links een oneven waarde tegenkomen en $A[j]$ ook oneven is (voor de best case) wordt j afgelaagd en blijft i gelijk en is dus $A[i]$ nog steeds oneven. Kortom, voor de best case zal vanaf het moment dat we de eerste oneven $A[i]$ tegenkomen $A[j]$ oneven moeten zijn. Vervolgens wordt in elke stap j 1 kleiner, totdat j gelijk is geworden aan i. Met andere woorden: het beste geval (dat zijn dus nul verwisselingen) treedt op als A er als volgt uitziet: een aantal —eventueel 0— even getallen, gevolgd door een aantal —eventueel 0— oneven getallen. Dit komt overeen met het worst case geval voor operatie I (zie **b.**).
In de worst case wordt in elke doorgang gewisseld, en dat kan alleen gebeuren als steeds $A[i]$ oneven is en $A[j]$ even. Vervolgens schuift steeds i naar rechts en j naar links. Er staan dus in de linkerhelft oneven getallen en in de rechterhelft even getallen. Als n even is ziet A er dus als volgt uit: $A[1]$ t/m $A[n/2]$ zijn oneven en $A[n/2+1]$ t/m $A[n]$ zijn even. Voor de worst case moeten immers alle $A[i]$ en $A[n-i+1]$ gewisseld worden, ook de middelste twee. Aantal verwisselingen: $n/2$. Als n oneven is kan er één element niet verwisseld worden, maar de rest wordt twee aan twee verwisseld. Worst case is in dit geval $\lfloor n/2 \rfloor$ verwisselingen. Dit komt voor als $A[1]$ t/m $A[\lceil n/2 \rceil]$ oneven zijn op één element na, en $A[\lceil n/2 \rceil + 1]$ t/m $A[n]$ even. Of als $A[1]$ t/m $A[\lfloor n/2 \rfloor]$ oneven zijn en $A[\lfloor n/2 \rfloor + 1]$ t/m $A[n]$ even op één element na.

d. Bij elke doorgang door de while wordt in elk geval de test in regel (3) gedaan. Alleen als die test False is wordt de test in regel (6) ook gedaan. Voor de worst case is ook het aantal doorgangen door de while-loop van belang. Het aantal keer dat II plaatsvindt is maximaal als het aantal rondes maximaal is *en* tevens het aantal tests per ronde. Dit maximale aantal kan ten duidelijkste bereikt worden, en wel (dan en slechts dan) als in elke ronde $A[i]$ oneven is *en* $A[j]$ oneven. In dat geval wordt alleen j met 1 afgelaagd, en blijft i op 1 staan. Er vinden dus $n-1$ rondes (doorgangen) plaats, met in elke ronde 2 vergelijkingen. In de laatste ronde is $i = 1, j = 2$ en mag $A[j]$ even zijn, want daarna stopt de while-loop altijd sowieso. Totaal aantal keer operatie II is in dat geval $2(n-1)$. Dit komt alleen voor als A geheel uit oneven getallen bestaat of als $A[2]$ even is en alle andere waarden in A oneven.